F(X) JOURNAL

Here is the second issue of the F(x) journal. As you will notice, we have phased out the question and answer section due to a lack of reader questions. We are also sorry for the late delivery of this issue. Again we are running into difficulties.

In this month's issue we have :

- Trig. Subroutines
- 6800 microprocessor : Part 2
- Conclusion

BUGS !! (sorry folks but were not perfect)

There was a bug in last month's program.
Line 310 should read :

```
310 K$ = KEY$(2) : IF K$ = "" GOTO 450
```

We are sorry for any problems this may have caused.

6800 - Your Microprocessor - Part 2

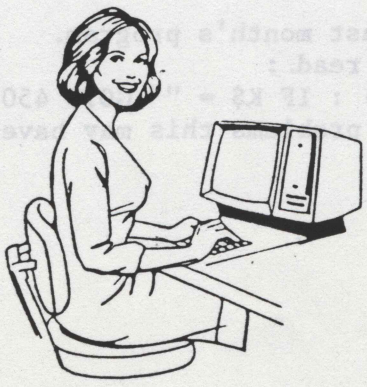
As was mentioned last month, accumulator A & B are used to perform simple arithmetic and data manipulation. Simple arithmetic means purely addition and subtraction. Since A & B can only store numbers from 0 to 255, in order to do addition or subtraction of larger numbers, individual bytes must be added or subtracted one at a time. When the result is greater than 255, the carry bit in the condition codes register will donate this. The carry bit is then figured in with the addition of the next higher-order byte.

In order to perform multiplication and division, the bit-shift instructions are used. In short, A & B are used as temporary storage places for data which must be manipulated or moved.

As was mentioned, the condition codes register is used to show the status of system after certain events. The ccr controls the following flag bits: V-overflow, C-carry, I-interrupt enable, S-sign, Z-zero, H- half-carry. The main bits are L,V,Z, and S. Those bits are essential to machine language branch instructions. This will be covered in a later article on machine language.

The remaining registers, X, SP, and PC are used for memory addressing. They can point at a memory location which contains data for A or B. This is most often done with the X register for an addition, this is called indexed addressing. There will be more on addressing modes in later articles. The stack pointer points at certain memory which is called the stack. The stack stores such things as data and memory addresses for later recall. When a subrotuine is called, the current address of the PC, which points at where the 6800 is currently executing instruction, is saved on the stack, the stack pointer is adjusted, and the subrotuine address is loaded into PC. When a return instruction is encountered, the old PC address is loaded back into the PC and the microprocessor continues.

The use of the registers and accumulators will be discussed next month in a multi-part series on machine language.



Last minute thoughts

1. We currently have the following programs available:
   A. BIORHYTHM - Calculate critical days !! ($4.95)
   B. TEXT STORAGE & RETREIVAL SYSTEM - File management system which is written in machine language. ($11.95)
   C. VOICE SYNTHESIZER - Software voice synthesizer. ($9.95)
   D. GRAPHIX PACKAGE - 3 utilities for high-res enthusiasts. ($7.95)
   E. MATH PROCESSOR - Performs many needed math function plus polynomial graphing. ($6.95)
   F. GAMES PACK - 4 intriguing games including Lord Logic Locks. ($8.95)

More programs will be available as soon as the documentation is complete.

If you would like to order an of the above programs or would like more information, please write us.

F(x) Software Co.
4246 Elisabeth Ave.
Holland, MI 49423

2. Next Month !!!

   A. More programs !!!
   B. Article on machine language
   C. Much, Much, More !!

COMPUTER PROGRAM

First we will give you the index and then we will give you the subrotuines. We advise that you input all of the parts of the subrotuine and use it as a block. This is because some of the Trig. functions use others in the subrotuine.

CALL ADDRESS	FUNCTION	INPUT	OUTPUT	VARIABLES USED
9020	Square Root	X	Y	W, Z
9140	Logarithms	X	L(Nat. Log) X(Com. Log)	T, U, V, W
9270	Sine	X(in degrees)	Y	Z
9440	Cosine	X(in degrees)	Y	W, Z, & Sine Rotuine
9510	Tangent	X(in degrees)	Y	T, V, W, Z & Sine & Cos. Rotuine

```
9000 REM * FUNCTIONS *
9010 REM
9020 REM # SQUARE ROOT #
9060 IF X=0 THEN Y=0 : RETURN
9070 IF X > 0 THEN 9090
9080 PRINT " ERROR - ROOT OF A NEGATIVE NUMBER ?" : STOP
9090 Y = X * .5 : Z = 0
9100 W = (X/Y - Y) * .5
9110 IF W = 0 THEN RETURN
9120 IF W = Z THEN RETURN
9130 Y = Y + W : Z = W : GOTO 9100
9140 REM # LOGARITHMS #
9200 W = 0 : IF X < 0 THEN PRINT " ERROR - LOG UNDEFINED AT "; X : STOP
9210 T = 1 : U = 2 : C = .5
9220 IF X >= T THEN X = V*X : W = W + T : GOTO 9220
9230 IF X < V THEN X = U*X : W = W - T : GOTO 9230
9240 X = (X - .7071)/(X + .7071) : L = X * X
9250 L = ((( .5989 * L + .9615) * L + 2.8854) * X + W - .5) * .6931
9260 X = L * .4343 : RETURN
9270 REM # SINE #
9320 IF X = 0 THEN Y = 0 : RETURN
9330 Z = ABS( X ) / X : X = Z * X
9340 IF X > 360 THEN X = X / 360 : X = (X - INT(X)) * 360
9350 IF X > 90 THEN X = X / 90 : Y = INT( Y ) : X = ( X - Y ) * 90 : ON
     Y GOTO 9380, 9390, 9400
9360 X = Y / 57.2958 : IF ABS( X )   .0002 THEN Y = 0 : RETURN
9370 GOTO 9410
9380 X = 90 - X : GOTO 9360
9390 X = -X : GOTO 9360
9400 X = X - 90 : GOT 9360
9410 Y = X - X * X * X / 6 + X * X * X * X * X / 120 - X * X * X * X * X * X
     * X / 5040
9420 Y = Y + X * X * X * X * X * X * X * X * X * X / 362880 : IF Z = -1 THEN
     Y = -Y
9430 RETURN
9440 REM # COSINE #
9490 W = ABS( X ) / X : X = X + 90 : GOSUB 9320 : IF Z = -1 IF W = 1
     THEN Y = -Y
9500 RETURN
9510 REM # TANGENT #
9560 T = X : GOSUB 9490
9570 V = Y : X = T : GOSUB 9320 : Y = Y / V
9580 RETURN
```